Software Fairness Dilemma: Is Bias Mitigation a Zero-Sum Game?

ZHENPENG CHEN, Nanyang Technological University, Singapore
XINYUE LI, Peking University, China
JIE M. ZHANG, King's College London, United Kingdom
WEISONG SUN*, Nanyang Technological University, Singapore
YING XIAO, King's College London, United Kingdom
TIANLIN LI, Nanyang Technological University, Singapore
YILING LOU, Fudan University, China
YANG LIU, Nanyang Technological University, Singapore

Fairness is a critical requirement for Machine Learning (ML) software, driving the development of numerous bias mitigation methods. Previous research has identified a leveling-down effect in bias mitigation for computer vision and natural language processing tasks, where fairness is achieved by lowering performance for all groups without benefiting the unprivileged group. However, it remains unclear whether this effect applies to bias mitigation for tabular data tasks, a key area in fairness research with significant real-world applications. This study evaluates eight bias mitigation methods for tabular data, including both widely used and cutting-edge approaches, across 44 tasks using five real-world datasets and four common ML models. Contrary to earlier findings, our results show that these methods operate in a zero-sum fashion, where improvements for unprivileged groups are related to reduced benefits for traditionally privileged groups. However, previous research indicates that the perception of a zero-sum trade-off might complicate the broader adoption of fairness policies. To explore alternatives, we investigate an approach that applies the state-of-the-art bias mitigation method solely to unprivileged groups, showing potential to enhance benefits of unprivileged groups without negatively affecting privileged groups or overall ML performance. Our study highlights potential pathways for achieving fairness improvements without zero-sum trade-offs, which could help advance the adoption of bias mitigation methods.

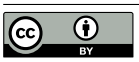
CCS Concepts: • **Software and its engineering → Software creation and management**; • **Computing methodologies → Machine learning**.

Additional Key Words and Phrases: Machine learning, software fairness, bias mitigation, sensitive attributes

ACM Reference Format:
Zhenpeng Chen, Xinyue Li, Jie M. Zhang, Weisong Sun, Ying Xiao, Tianlin Li, Yiling Lou, and Yang Liu. 2025. Software Fairness Dilemma: Is Bias Mitigation a Zero-Sum Game?. *Proc. ACM Softw. Eng.* 2, FSE, Article FSE080 (July 2025), 22 pages. https://doi.org/10.1145/3729350

*Corresponding author.

Authors' Contact Information: Zhenpeng Chen, Nanyang Technological University, Singapore, Singapore, zhenpeng.chen@ntu.edu.sg; Xinyue Li, Peking University, Beijing, China, xinyueli@stu.pku.edu.cn; Jie M. Zhang, King's College London, London, United Kingdom, jie.zhang@kcl.ac.uk; Weisong Sun, Nanyang Technological University, Singapore, Singapore, weisong.sun@ntu.edu.sg; Ying Xiao, King's College London, London, United Kingdom, ying.1.xiao@kcl.ac.uk; Tianlin Li, Nanyang Technological University, Singapore, Singapore, tianlin001@e.ntu.edu.sg; Yiling Lou, Fudan University, Shanghai, China, yilinglou@fudan.edu.cn; Yang Liu, Nanyang Technological University, Singapore, Singapore, yangliu@ntu.edu.sg.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2025 Copyright held by the owner/author(s).
ACM 2994-970X/2025/7-ARTFSE080
https://doi.org/10.1145/3729350

1 Introduction

Machine Learning (ML) software is increasingly adopted in critical decision-making domains, such as criminal sentencing, hiring, healthcare, and finance [10, 12, 39]. However, it frequently exhibits discrimination based on sensitive attributes such as sex, race, and age [14]. As a result, fairness has become a key requirement for ML software, drawing significant attention from software researchers and engineers [7, 19, 40]. In response, the Software Engineering (SE) community has invested considerable efforts into developing bias mitigation methods [2, 4, 9, 10, 14, 16, 24, 31, 35].

In the software fairness literature, "unfairness" typically measures performance disparities across demographic groups defined by sensitive attributes [12, 14, 39]. Bias mitigation methods aim to reduce these disparities, thereby improving fairness.

However, this approach is often criticized as "strictly egalitarian" [25, 28], because it emphasizes relative performance between groups while neglecting absolute performance. This can lead to a phenomenon known as "leveling down," where fairness is achieved by lowering performance for all groups without benefiting the unprivileged group [25, 28, 34, 43]. Zietlow et al. [43] highlight this issue in bias mitigation methods for Computer Vision (CV) tasks, showing that these methods degrade performance for all groups, with the most significant drop for the privileged group. Similarly, Maheshwari et al. [25] observe similar effects in CV and Natural Language Processing (NLP) tasks involving multiple sensitive attributes, where bias mitigation methods reduce the outcomes for privileged groups without benefiting unprivileged ones.

Despite these efforts, to the best of our knowledge, similar studies on tabular data tasks are lacking. Addressing this gap is important because bias mitigation methods for tabular data have critical applications in fairness-sensitive domains and represent the most extensively studied area in software fairness [20]. Existing empirical studies on bias mitigation in tabular data tasks primarily focus on trade-offs between fairness and overall performance [4, 13, 21]. In contrast, this paper aims to investigate trade-offs in group benefits.

We conduct a large-scale empirical study on eight representative bias mitigation methods for tabular data, including both widely used and cutting-edge techniques. We evaluate these methods across 44 tabular data tasks spanning social, financial, and medical domains, using five real-world datasets and four common ML models, while considering scenarios with both single and multiple sensitive attributes.

Unlike previous findings in CV and NLP [25, 43], our study reveals that bias mitigation methods for tabular data exhibit a zero-sum trade-off, which indicates that gains for one group result in corresponding losses for another [6, 30]. Specifically, we observe that improvements for unprivileged groups are accompanied by reductions in benefits for traditionally privileged groups. For example, existing bias mitigation methods significantly increase the selection rate (i.e., the rate of assigning favorable outcomes) and true positive rate for unprivileged groups by large effect sizes in 37.5% to 84.4% and 21.9% to 87.5% of tasks, respectively. Meanwhile, these methods cause significant reductions in the selection rate and true positive rate for privileged groups, with large effect sizes in 31.3% to 62.5% and 25.0% to 62.5% of tasks, respectively. Furthermore, greater fairness improvements are significantly associated with larger gains in the selection and true positive rates for unprivileged groups, but also with more substantial decreases in these rates for privileged groups.

However, previous research [6, 30] suggests that the perception of a zero-sum trade-off may hinder the broader adoption of fairness policies. This belief can pose a significant challenge, as even individuals with strong egalitarian values may resist fairness initiatives if they believe that improving outcomes for one group comes at the expense of another [6]. As a result, the zero-sum trade-offs observed in bias mitigation methods for tabular data not only present technical challenges

but may also exacerbate sociopolitical barriers, potentially limiting the widespread acceptance of these methods.

This motivates us to explore whether it is possible to avoid a zero-sum trade-off in bias mitigation methods for tabular data, thereby facilitating the adoption of these methods in practice. To this end, we investigate applying the state-of-the-art bias mitigation method solely to unprivileged groups. This approach enhances benefits for unprivileged groups while preserving the benefits for privileged groups and maintaining overall ML performance (e.g., accuracy and F1-score) comparable to applying the method universally. Additionally, the approach results in a marginal increase in the overall selection rate, averaging just 0.01, compared to applying the method across the entire population. This study serves as a preliminary exploration of pathways toward achieving fairness improvements without incurring zero-sum trade-offs.

In summary, this paper makes the following contributions:

- We conduct a large-scale empirical study to characterize the impact of existing bias mitigation methods for tabular data on different demographic groups.
- We perform a preliminary investigation into the possibility of avoiding the zero-sum trade-off in bias mitigation for tabular data.
- We provide a replication package [1] that includes all data and code used in the paper to support replication and further research.

2 Preliminaries

We begin by providing background knowledge on ML software fairness and discussing the relevance of bias mitigation to SE, followed by a review of related work.

2.1 ML Software Fairness

We focus on ML classification for tabular data, the most widely studied area in software fairness research [4, 5, 9, 10, 12, 14, 21, 31, 33, 35, 39], particularly because of its crucial role in fields such as finance, healthcare, and criminal justice, where fairness is essential. Decision-making involving tabular data often includes explicit sensitive attributes such as race, sex, and age, which, if mishandled, can lead to discriminatory outcomes.

Sensitive attributes can create divisions between privileged and unprivileged groups. In practice, ML software often favors privileged groups by assigning them positive labels, while unprivileged groups receive unfavorable labels. For example, an income prediction system might be more likely to predict high income for males (privileged group) and low income for females (unprivileged group), with sex as the sensitive attribute, high income as the favorable label, and low income as the unfavorable label.

Such bias has driven extensive research into bias mitigation to improve the group fairness of ML software [9, 10, 12, 31, 35], which is also the focus of our paper. Group fairness, widely encoded in laws and regulations, is recognized as a crucial non-functional requirement of ML software [11]. It ensures that different demographic groups, defined by sex, race, age, or other sensitive attributes, are treated equitably by ML software, preventing biased decisions that favor one group over others.

To quantitatively measure group fairness, various metrics have been proposed, including Statistical Parity Difference (SPD), Equal Opportunity Difference (EOD), and Average Odds Difference (AOD), all of which are widely adopted in both literature and practice [20]. These metrics share a common motivation: to balance classification performance across different demographic groups.

Let A denote a sensitive attribute, where 1 represents the privileged group and 0 the unprivileged group. Y represents the actual label, and $\hat{Y}$ the predicted label, with 1 indicating the favorable label and 0 the unfavorable label. Below are the definitions and calculations of these fairness metrics:

SPD measures the difference in selection rates between the privileged and unprivileged groups.

$$SPD = |P[\hat{Y} = 1|A = 1] - P[\hat{Y} = 1|A = 0]|. \tag{1}$$

EOD measures the difference in true positive rates between the privileged and unprivileged groups.

$$EOD = |P[\hat{Y} = 1|A = 1, Y = 1] - P[\hat{Y} = 1|A = 0, Y = 1]|. \tag{2}$$

AOD measures the average of the differences in true positive rates and false positive rates between the privileged and unprivileged groups.

$$AOD = |\frac{1}{2}[(P[\hat{Y} = 1|A = 1, Y = 1] - P[\hat{Y} = 1|A = 0, Y = 1]) + (P[\hat{Y} = 1|A = 1, Y = 0] - P[\hat{Y} = 1|A = 0, Y = 0])]|. \tag{3}$$

To extend these fairness metrics to scenarios with multiple sensitive attributes, we divide the population into multiple demographic groups based on the sensitive attributes considered. For example, with two sensitive attributes, i.e., sex (male, female) and race (white, non-white), we generate four groups: white male, white female, non-white male, and non-white female. We then quantify the maximum disparity between these groups by measuring the difference among the groups with the least and greatest discrimination [14]. Specifically, for SPD, we calculate the maximum difference in selection rates among the groups; for EOD, the maximum difference in true positive rates; and for AOD, the maximum average difference in true positive rates and false positive rates. Due to the page limit, we omit the detailed equations.

2.2 Relevance of Bias Mitigation to SE

Bias mitigation is a highly relevant topic to SE, holding significant importance for software researchers, engineers, and companies. **(1) For software researchers:** Fairness is a critical non-functional software requirement [7, 19, 40], and unfairness in ML software is regarded as a fairness bug by the SE community [11, 12]. Thus, bias mitigation (i.e., addressing fairness bugs) has received substantial attention from SE researchers. Five of the bias mitigation methods we study (Section 3.2) were introduced in top SE venues (ICSE, FSE, and TSE). Notably, FairSMOTE [9], one of these methods, earned the Distinguished Paper Award at FSE'21, underscoring its impact and relevance to SE. **(2) For software engineers:** Ensuring fairness in ML software is widely recognized as an ethical duty for software engineers [9, 14], particularly as they integrate ML models into software systems while adhering to ethical and legal standards. **(3) For software companies:** Unfair ML software poses significant risks to software companies, including ethical, reputational, financial, and legal consequences, particularly if they violate anti-discrimination laws [13, 39].

2.3 Related Work

Bias Mitigation Methods. Numerous bias mitigation methods have been proposed to enhance group fairness. These methods are generally categorized into three types: pre-processing, in-processing, and post-processing [20]. (1) *Pre-processing methods* reduce bias in the training data, thereby improving the fairness of the resulting ML models. For instance, Chakraborty et al. [9] identified bias in prior decisions as a root cause of ML software bias, and addressed this by rebalancing internal data distributions and removing biased labels from the training data. (2) *In-processing methods* enhance fairness during the model training process. For example, Gao et al. [16] proposed a method specifically for deep neural networks, which involves detecting neurons that exhibit contradictory optimization directions for accuracy and fairness, followed by selective dropout training to mitigate bias. (3) *Post-processing methods* adjust the model's outcomes to ensure fairer

results. For example, Peng et al. [31] modified the values of sensitive attributes in the test data to make the model's decisions fair. There are also hybrid approaches that span different bias mitigation types. For example, Xiao et al. [35] proposed MirrorFair, which first processes the training dataset to generate a counterfactual dataset, then trains models separately on both datasets, and finally adaptively combines their predictions to adjust the model's outcomes. Works like MirrorFair [35] present technical contributions by proposing new bias mitigation methods. In contrast, our paper makes an empirical contribution by systematically examining the zero-sum trade-offs between groups during bias mitigation, an aspect not addressed in these papers.

Empirical Evaluation. To systematically understand the effects of bias mitigation methods, researchers have conducted numerous empirical investigations. Chen et al. [14] identified a side effect of bias mitigation: addressing bias concerning one sensitive attribute can inadvertently amplify bias regarding others. Biswas and Rajan [4] applied bias mitigation techniques to ML models sourced from Kaggle, a crowd-sourced platform, to analyze their impact on fairness. Recognizing that fairness improvements often come at the cost of ML performance, they also assessed the impact of these methods on ML performance. Hort et al. [21] and Chen et al. [13] conducted studies to characterize the fairness-performance trade-off achieved by existing bias mitigation methods. Similarly, Friedler et al. [15] and Menon and Williamson [27] studied the trade-offs between fairness and overall accuracy (an important ML performance metric).

All these studies focus on bias mitigation for tabular data, which is the most extensively studied in the software fairness literature [20]. In contrast, Yang et al. [36] conducted an empirical study of bias mitigation methods specifically for image classification tasks, evaluating their effects on both fairness and ML performance.

These empirical studies primarily focus on overall performance without providing a more in-depth analysis of how performance varies across different demographic groups. To address this gap, a few studies have been conducted. Zietlow et al. [43] evaluated bias mitigation methods in CV tasks and found that these methods improve fairness by degrading performance across all groups. Similarly, Maheshwari et al. [25] conducted an empirical study on bias mitigation methods for CV and NLP tasks involving multiple sensitive attributes, noting that these methods improve fairness by harming the best-off group without benefiting the worst-off group. In contrast, there is a lack of similar empirical studies on tabular data, despite its prominence in fairness research. Zafar et al. [37] introduced a fairness measure related to disparate mistreatment and explored its implications for group benefits, but they did not evaluate the trade-offs in group benefits achieved by existing bias mitigation methods.

3 Experimental Setup

This section describes our research questions and experimental setup.

3.1 Research Questions (RQs)

We aim to answer the following RQs in this study.

RQ1: *How do bias mitigation methods impact various demographic groups in tabular data tasks?* Existing studies have addressed this question on CV and NLP tasks [25, 43], but this paper shifts the focus to tabular data, which is the most extensively studied topic in software fairness research [20].

RQ2: *What is the correlation between the impact on different demographic groups and the overall impact on fairness?* Since bias mitigation methods aim to enhance fairness, it is important to quantitatively assess how changes in fairness correlate with the effects on different demographic groups. This analysis can provide deeper insights into the relationship between fairness improvements and group-specific outcomes.

RQ3: *How do bias mitigation methods impact demographic groups defined by multiple sensitive attributes?* As research increasingly considers multiple sensitive attributes simultaneously [14], the complexity of assessing bias mitigation methods grows. This RQ explores how bias mitigation methods for tabular data perform on each group when multiple attributes are considered, leading to a more nuanced understanding of their effects across intersecting demographic groups.
RQ4: *Is it possible to improve fairness without degrading the performance of either privileged or unprivileged groups?* Bias mitigation methods that do not lead to negative outcomes for any group are likely to be more widely adopted by society. This question investigates whether such an ideal situation is achievable and explores alternative approaches that might allow for fairness improvements without detriment to any group.

3.2 Bias Mitigation Methods

We use a total of eight existing bias mitigation methods for our study. On one hand, we employ three most widely adopted bias mitigation methods, as identified in a recent survey [20]: Adversarial Debiasing (ADV) [38], Reweighting (REW) [22], and Equalized Odds Processing (EOP) [18]. On the other hand, we include five recently proposed methods from the software fairness literature: FairSMOTE [9], LTDD [24], MAAT [12], FairMask [31], and MirrorFair [35]. Our selection spans pre-processing, in-processing, and post-processing approaches, ensuring a comprehensive evaluation.

In the following, we provide a brief introduction to each method.

- ADV [38] uses adversarial learning to make predictions less dependent on sensitive attributes during the training process.
- REW [22] adjusts the weights of different instances in the training data to ensure that underrepresented groups are given more importance.
- EOP [18] adjusts model predictions to ensure that the false positive and false negative rates are similar across different demographic groups.
- FairSMOTE [9] balances internal distributions of training data concerning both sensitive attributes and class labels, while also removing biased labels.
- LTDD [24] is a data debugging method that identifies and excludes biased parts of features in the training data for building fair ML software.
- MAAT [12] trains individual models optimized separately for ML performance and fairness, subsequently combining their predictions for a balanced fairness-performance trade-off.
- FairMask [31] trains extrapolation models to predict sensitive attributes based on other features and uses these models to relabel sensitive attributes during the inference process.
- MirrorFair [35] constructs a counterfactual dataset from the original data, trains models on both datasets, and adaptively combines their predictions for fair decisions.

3.3 Bias Mitigation Tasks

We use the same set of 44 bias mitigation tasks as previous work [35], which are generated using five real-world datasets and four widely used ML models.

Datasets. Table 1 presents an overview of the datasets used in our study. Specifically, we use five real-world tabular datasets extensively employed in fairness research [12, 13, 35, 39], spanning different domains and fairness-critical scenarios: income prediction, recidivism prediction, credit prediction, deposit subscription prediction, and healthcare needs prediction. These datasets span the financial, social, and medical domains, providing a broad scope for analysis.

For each dataset, we use the sensitive attributes as defined within it. These datasets cover sex, race, and age, which are recognized as the three most commonly considered sensitive attributes [20].

Table 1. Datasets.

Name	Size	Sensitive attr(s)	Favorable label	Description
Adult	45,222	sex, race	income > 50K	predict whether an individual's income exceeds 50K
Compas	6,167	sex, race	no recidivism	predict recidivism for criminal defendants
German	1,000	sex, age	good credit	predict whether an individual has good credit
Bank	30,488	age	subscriber	predict whether an individual will subscribe to a deposit
Mep	15,830	race	utilizer	predict individual healthcare needs

We follow previous work [35] to transform the five datasets into 11 tasks: 8 single-attribute tasks (Adult-Sex, Adult-Race, Compas-Sex, Compas-Race, German-Sex, German-Age, Bank-Age, and Mep-Race) and 3 multiple-attribute tasks (Adult-Sex-Race, Compas-Sex-Race, and German-Sex-Age).

ML Models. For each task, we train four representative ML models: Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN). These models are demonstrated to be the most extensively studied in fairness research [20] and widely used in fairness-critical decision applications [14]. For LR, RF, and SVM, we use the configurations specified in recent studies [12–14, 21]. For DNN, we employ a fully connected network with five hidden layers comprising 64, 32, 16, 8, and 4 units, respectively, a model architecture extensively used for the datasets we study [13, 14, 41, 42].

By converting the datasets into eight single-attribute tasks and three multiple-attribute tasks, and training four models for each, we finally obtain $8 \times 4 = 32$ single-attribute bias mitigation tasks and $3 \times 4 = 12$ multiple-attribute bias mitigation tasks.

3.4 Evaluation Metrics

We use the same set of fairness metrics as in previous work [12, 14], including SPD, EOD, and AOD. The selection of fairness metrics aligns with established practices in the fairness literature, covering the most widely adopted metrics in the field according to a recent survey [20]. Detailed descriptions of these metrics are provided in Section 2.

To explore the zero-sum trade-offs between the benefits for privileged and unprivileged groups, we disaggregate the fairness metrics by focusing on three group-level performance metrics: selection rate (SR), true positive rate (TPR), and false positive rate (FPR). These performance metrics are used by the fairness metrics to measure the benefits for each group. Higher SR and TPR values indicate better performance and greater benefits, while a lower FPR value typically suggests better performance. However, it is important to note that a high FPR for a demographic group does not mean the group is being harmed. In bias mitigation tasks, the positive label represents a favorable outcome, so a high FPR indicates that the bias mitigation method tends to favor this group, even assigning favorable outcomes to the group members who are not qualified.

For single-attribute tasks, we denote these rates for the privileged group as SR_P, TPR_P, and FPR_P, and for the unprivileged group as SR_U, TPR_U, and FPR_U. In multiple-attribute tasks, demographic groups are ranked based on the proportion of members achieving favorable outcomes in the training data, producing groups labeled $Group_1$, $Group_2$, ..., $Group_n$, where $Group_1$ is the most favored and $Group_n$ is the least favored. The performance for $Group_i$ is denoted by SR_i, TPR_i, and FPR_i.

3.5 Implementation Details

To ensure the reproducibility of our results, we outline the implementation details. Consistent with previous work [14], each bias mitigation method is applied to each task 20 times, with performance and fairness metrics averaged across these runs to mitigate the impact of randomness. In each experiment, the dataset is randomly split, with 70% used as training data and 30% as test data, following a common practice in the software fairness literature [12, 14, 35].

4 RQ1: Impact on Different Demographic Groups

In this RQ, we apply bias mitigation methods to 32 single-attribute bias mitigation tasks. In such tasks, the population is typically divided into two groups: a privileged group and an unprivileged group. We examine the SR, TPR, and FPR of the two groups, both with and without the application of bias mitigation methods, to assess their impact on the performance for different groups. Specifically, we analyze the frequency of various impact types (e.g., performance decrease or increase) across the 32 tasks (**RQ1.1**), followed by an analysis of the effect size of these impacts (**RQ1.2**).

4.1 RQ1.1: Frequency Analysis

Methodology. To determine whether the performance for a group is significantly impacted by a bias mitigation method, we use the non-parametric Mann-Whitney U-test [26], a widely adopted approach in software fairness research [13, 14, 35]. Following established practices [13, 14, 35], we consider an impact to be statistically significant only if the p-value from the test is below 0.05, ensuring our findings have a 95% confidence level. For example, when comparing two sets of SR values for the unprivileged group across 20 runs, with and without the bias mitigation method, the null hypothesis assumes no significant difference in SR between the two conditions. If the test yields a p-value < 0.05, we can reject the null hypothesis and conclude with 95% confidence that applying the bias mitigation method significantly affects the SR of the unprivileged group.

We classify impact types as follows: an increase in the average metric value with a p-value < 0.05 is considered a significant increase; a decrease with a p-value < 0.05 is considered a significant decrease; and if the p-value $\geq$ 0.05, the result is considered a tie. For each bias mitigation method, we calculate the number of scenarios in which it leads to a significant increase, tie, or significant decrease in each group performance metric.

Results. Table 2 presents the results. Overall, bias mitigation methods tend to decrease SR, TPR, and FPR for the privileged group while increasing them for the unprivileged group, thereby narrowing the performance gap and improving fairness. Initially, the original models show higher SR, TPR, and FPR for the privileged group compared to the unprivileged group. The bias mitigation methods we study significantly reduce SR_P, TPR_P, and FPR_P in 52.0% (133/256), 50.8% (130/256), and 45.7% (117/256) of tasks, respectively, while significantly increasing them in 7.4% (19/256), 7.4% (19/256), and 6.6% (17/256) of tasks. In contrast, these methods significantly increase SR_U, TPR_U, and FPR_U in 64.1% (164/256), 55.5% (142/256), and 60.9% (156/256) of tasks, while significantly decreasing them in 3.1% (8/256), 4.3% (11/256), and 2.7% (7/256) of tasks.

This pattern holds across all eight methods studied. Each method more frequently decreases SR_P, TPR_P, and FPR_P than increases them, while it more frequently increases SR_U, TPR_U, and FPR_U than decreases them. For instance, the state-of-the-art MirrorFair decreases SR_P in 46.9% (15/32) of tasks but increases it in only 6.3% (2/32). Conversely, these methods are more likely to increase SR_U, TPR_U, and FPR_U. For example, MirrorFair increases SR_U in 87.5% (28/32) of tasks, with a decrease in just 6.3% (2/32).

In summary, we present two key observations.

- Unlike previous bias mitigation studies in CV and NLP, where fairness improvements typically result from reducing performance for both groups (with a larger reduction for the privileged group), we do not observe the same leveling-down effect in tabular data tasks. Instead, bias mitigation methods for tabular data exhibit a zero-sum trade-off, where they improve SR, TPR, and FPR for the unprivileged group while reducing them for the privileged group. This zero-sum pattern is consistent across all eight methods and all three metrics (SR, TPR, and FPR).
- The choice of performance metric is critical. Since higher SR and TPR, and lower FPR, generally indicate better performance, we conclude that current bias mitigation methods for tabular

Table 2. (RQ1.1) Number of tasks where each bias mitigation method significantly increases (↑), decreases (↓), or does not significantly impact (–) the SR, TPR, and FPR for each group. Overall, bias mitigation methods for tabular data tend to decrease SR, TPR, and FPR for the privileged group while increasing them for the unprivileged group.

Method	SR_P (↑)	SR_P (–)	SR_P (↓)	SR_U (↑)	SR_U (–)	SR_U (↓)
ADV	4	14	14	16	12	4
REW	0	10	22	21	11	0
EOP	1	14	17	22	10	0
FairSMOTE	8	4	20	24	8	0
LTDD	2	15	15	17	13	2
MAAT	2	11	19	21	11	0
FairMask	0	21	11	15	17	0
MirrorFair	2	15	15	28	2	2
Overall	19	104	133	164	84	8
Method	TPR_P (↑)	TPR_P (–)	TPR_P (↓)	TPR_U (↑)	TPR_U (–)	TPR_U (↓)
ADV	4	11	17	16	12	4
REW	0	12	20	18	14	0
EOP	1	12	19	14	15	3
FairSMOTE	9	2	21	21	11	0
LTDD	1	17	14	16	14	2
MAAT	2	14	16	19	13	0
FairMask	0	22	10	10	22	0
MirrorFair	2	17	13	28	2	2
Overall	19	107	130	142	103	11
Method	FPR_P (↑)	FPR_P (–)	FPR_P (↓)	FPR_U (↑)	FPR_U (–)	FPR_U (↓)
ADV	3	18	11	15	13	4
REW	0	12	20	20	12	0
EOP	2	19	11	22	10	0
FairSMOTE	8	6	18	23	9	0
LTDD	2	16	14	17	14	1
MAAT	2	12	18	20	12	0
FairMask	0	21	11	12	20	0
MirrorFair	0	18	14	27	3	2
Overall	17	122	117	156	93	7

data improve SR and TPR for the unprivileged group at the expense of the privileged group, but decrease the unprivileged group's performance in terms of FPR. However, as discussed in Section 3.4, higher FPR can also indicate greater benefits in bias mitigation contexts. Thus, the overall effect of these methods is to promote fairness by enhancing benefits for the unprivileged group, even if this comes at a cost to the privileged group.

Finding 1: Contrary to previous findings in CV and NLP, where bias mitigation methods typically improve fairness by reducing performance for both groups (a phenomenon known as leveling down), our study on tabular data tasks reveals a zero-sum trade-off. Specifically, we find that bias mitigation methods tend to adjust the selection rate, true positive rate, and false positive rate by lowering these metrics for the privileged group and raising them for the unprivileged group, thus narrowing disparities and enhancing fairness. Notably, this pattern holds consistently across all eight methods we examine.

4.2 RQ1.2: Effect Size Analysis

Methodology. To further analyze the impact of bias mitigation methods on each group, we employ Cliff's δ [23], a widely used effect size metric in SE research [3, 14, 23]. Following standard practice [3, 14, 23], we interpret a δ value with an absolute magnitude of 0.428 or greater as indicating a large effect size. First, we determine the proportion of tasks where each method significantly decreases SR, TPR, and FPR by a large effect size. Second, for each method, we calculate the mean

Table 3. (RQ1.2) Effect size of bias mitigation methods on performance for each group. The table presents the mean impact and the maximum performance decrease/increase for the privileged/unprivileged groups, along with how these relative changes are derived from absolute numbers. For example, the mean impact of ADV in SR_P is -0.026 (0.462-0.489), indicating that ADV reduces the average SR_P across tasks from 0.489 to 0.462. The table also shows the proportions of tasks where each method results in a significantly large performance decrease/increase for these groups. We find that existing bias mitigation methods substantially increase the benefits for the unprivileged group, with significant increases in SR_U and TPR_U in 37.5%~84.4% and 21.9%~87.5% of tasks, respectively. However, these methods also lead to significant reductions in the SR_P and TPR_P, with large effect sizes observed in 31.3%~62.5% and 25.0%~62.5% of tasks, respectively.

Method	SR_P			TPR_P			FPR_P		
	Mean	Max ↓	Large ↓	Mean	Max ↓	Large ↓	Mean	Max ↓	Large ↓
ADV	-0.026 (0.462-0.489)	-0.126 (0.716-0.841)	43.8%	-0.049 (0.654-0.704)	-0.175 (0.460-0.635)	50.0%	0.000 (0.334-0.334)	-0.085 (0.497-0.582)	34.4%
REW	-0.043 (0.446-0.489)	-0.177 (0.651-0.828)	59.4%	-0.058 (0.646-0.704)	-0.189 (0.350-0.539)	56.3%	-0.043 (0.291-0.334)	-0.213 (0.476-0.690)	56.3%
EOP	-0.024 (0.465-0.489)	-0.110 (0.698-0.808)	37.5%	-0.043 (0.660-0.704)	-0.162 (0.320-0.482)	46.9%	-0.011 (0.323-0.334)	-0.075 (0.493-0.568)	34.4%
FairSMOTE	-0.026 (0.462-0.489)	-0.156 (0.672-0.828)	62.5%	-0.035 (0.669-0.704)	-0.357 (0.182-0.539)	62.5%	-0.035 (0.299-0.334)	-0.180 (0.392-0.572)	56.3%
LTDD	-0.033 (0.456-0.489)	-0.212 (0.616-0.828)	40.6%	-0.044 (0.659-0.704)	-0.219 (0.320-0.539)	40.6%	-0.032 (0.303-0.334)	-0.245 (0.445-0.690)	37.5%
MAAT	-0.025 (0.464-0.489)	-0.078 (0.196-0.273)	56.3%	-0.043 (0.660-0.704)	-0.138 (0.477-0.615)	50.0%	-0.019 (0.315-0.334)	-0.051 (0.058-0.109)	53.1%
FairMask	-0.015 (0.474-0.489)	-0.062 (0.766-0.828)	31.3%	-0.014 (0.689-0.704)	-0.056 (0.722-0.778)	25.0%	-0.020 (0.314-0.334)	-0.093 (0.597-0.690)	21.9%
MirrorFair	-0.015 (0.474-0.489)	-0.063 (0.764-0.828)	46.9%	-0.018 (0.685-0.704)	-0.098 (0.517-0.615)	37.5%	-0.017 (0.317-0.334)	-0.080 (0.610-0.690)	43.8%
Method	SR_U			TPR_U			FPR_U		
	Mean	Max ↑	Large ↑	Mean	Max ↑	Large ↑	Mean	Max ↑	Large ↑
ADV	0.006 (0.354-0.348)	0.081 (0.198-0.116)	50.0%	0.043 (0.649-0.605)	0.226 (0.719-0.493)	46.9%	0.006 (0.235-0.229)	0.057 (0.096-0.039)	46.9%
REW	0.036 (0.385-0.348)	0.144 (0.750-0.606)	62.5%	0.055 (0.661-0.605)	0.187 (0.680-0.493)	56.3%	0.038 (0.267-0.229)	0.180 (0.613-0.433)	56.3%
EOP	0.044 (0.392-0.348)	0.160 (0.721-0.562)	62.5%	0.025 (0.631-0.605)	0.105 (0.826-0.721)	40.6%	0.057 (0.287-0.229)	0.221 (0.612-0.391)	68.8%
FairSMOTE	0.049 (0.397-0.348)	0.250 (0.856-0.606)	75.0%	0.081 (0.687-0.605)	0.310 (0.685-0.375)	62.5%	0.050 (0.279-0.229)	0.312 (0.745-0.433)	68.8%
LTDD	0.031 (0.379-0.348)	0.182 (0.788-0.606)	53.1%	0.044 (0.649-0.605)	0.205 (0.725-0.520)	43.8%	0.034 (0.263-0.229)	0.213 (0.646-0.433)	46.9%
MAAT	0.031 (0.379-0.348)	0.151 (0.755-0.605)	62.5%	0.035 (0.640-0.605)	0.130 (0.865-0.736)	46.9%	0.035 (0.264-0.229)	0.178 (0.610-0.433)	59.4%
FairMask	0.014 (0.362-0.348)	0.060 (0.622-0.562)	37.5%	0.015 (0.621-0.605)	0.064 (0.785-0.721)	21.9%	0.016 (0.246-0.229)	0.101 (0.611-0.509)	25.0%
MirrorFair	0.063 (0.411-0.348)	0.241 (0.846-0.605)	84.4%	0.072 (0.677-0.605)	0.208 (0.944-0.736)	87.5%	0.071 (0.300-0.229)	0.286 (0.718-0.433)	81.3%

and maximum decrease in SR_P, TPR_P, and FPR_P across 32 single-attribute tasks. We perform a similar analysis to assess the effect size of decreases for SR_U, TPR_U, and FPR_U.

Results. Table 3 presents the results. The table presents the mean impact (the "Mean" columns) and the maximum decrease/increase in performance for the privileged/unprivileged group (the "Max ↓" and "Max ↑" columns), along with how these relative changes are derived from absolute numbers. For example, the mean impact of ADV in SR_P is -0.026 (0.462-0.489), indicating that ADV reduces the average SR_P across tasks from 0.489 to 0.462. The table also shows the proportions of tasks where each method results in a significant decrease/increase in performance by a large effect size for the privileged/unprivileged groups (the "Large ↓" and "Large ↑" columns).

First, we find that existing methods substantially increase the benefits for the unprivileged group. Specifically, these methods significantly increase SR_U, TPR_U, and FPR_U by a large effect size in 37.5%~84.4%, 21.9%~87.5%, and 25.0%~81.3% of tasks, respectively. In terms of the proportion of tasks with a significant increase in benefits for the unprivileged group, MirrorFair ranks first, followed by FairSMOTE.

Second, we observe that existing bias mitigation methods generally reduce the favorable outcomes for the privileged group, with significant decreases in SR_P, TPR_P, and FPR_P by a large effect size in 31.3%∼62.5%, 25.0%∼62.5%, and 21.9%∼56.3% of tasks, respectively.

While these methods are effective in narrowing disparities, the reduction in favorable outcomes for the privileged group may raise concerns regarding social perceptions. Research suggests that members of the privileged group may perceive fairness efforts as introducing bias against them [6, 30], and thus resist fairness policies. If bias mitigation methods are perceived as disproportionately affecting the privileged group, this could result in resistance to their broader adoption, potentially complicating efforts to achieve widespread fairness. Addressing this challenge will be essential in promoting the successful implementation of bias mitigation strategies.

Finding 2: Existing bias mitigation methods substantially increase the benefits for the unprivileged group, with significant increases in the selection rate and true positive rate by a large effect size in 37.5%∼84.4% and 21.9%∼87.5% of tasks, respectively. However, these methods also lead to significant reductions in the selection rate and true positive rate for the privileged group, with large effect sizes observed in 31.3%∼62.5% and 25.0%∼62.5% of tasks, respectively. While these reductions aim to narrow disparities and promote fairness, they may also contribute to the perception that efforts to reduce bias against the unprivileged group introduce bias against the privileged group. Such perceptions could create tension and potentially hinder the wider adoption of bias mitigation strategies, complicating efforts to achieve equitable outcomes.

5 RQ2: Correlation Between Group Impact and Fairness

Methodology. To explore the relationship between the impact on each group and fairness, we first calculate the changes in values for SR_P, TPR_P, FPR_P, SR_U, TPR_U, FPR_U, SPD, EOD, and AOD for each (bias mitigation method, task) pair by subtracting the original value from the value after applying the method. This generates a list of 256 value changes for each of the nine metrics, corresponding to the eight methods applied across 32 single-attribute bias mitigation tasks.

We then compute Spearman's correlation coefficient [29] between these lists for each pair of metrics. The coefficient ρ ranges from -1 to 1, where 1 indicates a perfect positive correlation, 0 indicates no correlation, and -1 indicates a perfect negative correlation. A correlation is considered statistically significant when the p-value is below 0.05 [14].

Results. Figure 1 illustrates the correlation coefficients. All presented correlations are statistically significant except for the two cases marked with ⊗.

We observe that the three fairness metrics (i.e., SPD, EOD, and AOD) are positively correlated, consistent with previous empirical findings [4, 13, 36], thereby validating the reliability of our correlation analysis. As a result, each fairness metric shows the same correlation trend with SR_P, TPR_P, FPR_P, SR_U, TPR_U, and FPR_U. Specifically, changes in SPD, EOD, and AOD are significantly positively correlated with SR_P, TPR_P, and FPR_P, and significantly negatively correlated with SR_U, TPR_U, and FPR_U. This indicates that greater fairness improvements, reflected by decreases in SPD, AOD, and EOD values, are associated with adjustments in SR, TPR, and FPR for both groups. In particular, fairness improvements are related to reductions in these metrics for the privileged group and corresponding increases for the unprivileged group, reflecting the narrowing of disparities.

Finding 3: In our study on bias mitigation methods for tabular data, greater fairness improvements are significantly associated with larger decreases in the selection rate, true positive

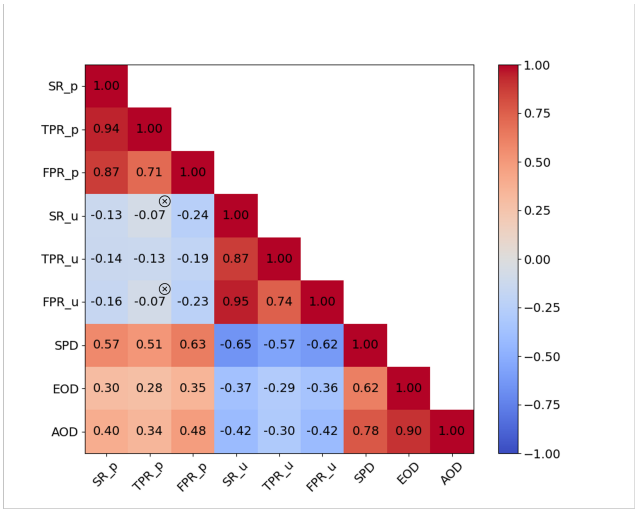


Fig. 1. (RQ2) Spearman's correlation across different metrics. The results reveal that greater fairness improvements, indicated by decreases in SPD, AOD, and EOD values, correspond to larger decreases in SR_P, TPR_P, and FPR_P, and larger increases in SR_U, TPR_U, and FPR_U.

> rate, and false positive rate for the privileged group, and larger increases in these rates for the unprivileged group.

6 RQ3: Impact With Multiple Sensitive Attributes

RQ3 explores whether the zero-sum pattern observed in RQ1 also applies to multiple-attribute tasks and examines how existing bias mitigation methods affect different demographic groups defined by multiple sensitive attributes.

Methodology. We consider 12 multiple-attribute tasks described in Section 3.3. For each task, the population is divided into four demographic groups, as each of the two sensitive attributes creates a division between privileged and unprivileged groups. The four groups are ranked based on the proportion of members achieving favorable outcomes in the training data, resulting in $Group_1$, $Group_2$, $Group_3$, and $Group_4$. $Group_1$ represents the most privileged group, while $Group_4$ represents the least privileged. We employ the Mann-Whitney U-test, as described in Section 4, to analyze the impact of existing methods on each of the four groups. Since LTDD does not support handling multiple sensitive attributes, we consider the remaining seven bias mitigation methods for answering RQ3.

Results. Table 4 presents the results. Due to space constraints, we display only the number of tasks where each method significantly increases or decreases performance on each group. Overall, we find that the zero-sum pattern also applies to multiple-attribute tasks.

First, bias mitigation methods tend to decrease the SR, TPR, and FPR for the most privileged group ($Group_1$). Specifically, these methods significantly reduce SR, TPR, and FPR for $Group_1$ in 48.8% (41/84), 46.4% (39/84), and 40.5% (34/84) of tasks, respectively. Notably, only the EOP method significantly decreases FPR for $Group_1$ in 2 tasks, but it significantly increases FPR in 3 tasks. In all other scenarios, each method significantly decreases SR, TPR, and FPR for $Group_1$ in at least as many tasks as it increases them.

Table 4. (RQ3) Number of tasks where each bias mitigation method significantly increases (↑) or decreases (↓) the SR, TPR, and FPR for each group when handling multiple sensitive attributes. Overall, bias mitigation methods for tabular data tend to decrease SR, TPR, and FPR for the most privileged group while increasing them for the least privileged group.

Method	SR_1 (↑)	SR_1 (↓)	SR_2 (↑)	SR_2 (↓)	SR_3 (↑)	SR_3 (↓)	SR_4 (↑)	SR_4 (↓)
ADV	3	5	2	7	6	1	4	3
REW	0	6	4	2	3	3	6	0
EOP	2	3	4	1	7	0	8	0
FairSMOTE	4	8	4	2	5	2	10	0
MAAT	4	4	3	5	6	4	9	1
FairMask	0	8	2	3	3	0	8	0
MirrorFair	2	7	4	5	8	2	12	0
Overall	15	41	23	25	38	12	57	4
Method	TPR_1 (↑)	TPR_1 (↓)	TPR_2 (↑)	TPR_2 (↓)	TPR_3 (↑)	TPR_3 (↓)	TPR_4 (↑)	TPR_4 (↓)
ADV	4	6	2	6	6	1	4	3
REW	0	6	3	2	3	3	5	0
EOP	2	3	3	2	5	1	3	0
FairSMOTE	4	7	3	2	6	2	8	0
MAAT	4	4	3	5	4	3	6	1
FairMask	0	7	1	2	3	0	5	0
MirrorFair	2	6	4	4	8	2	12	0
Overall	16	39	19	23	35	12	43	4
Method	FPR_1 (↑)	FPR_1 (↓)	FPR_2 (↑)	FPR_2 (↓)	FPR_3 (↑)	FPR_3 (↓)	FPR_4 (↑)	FPR_4 (↓)
ADV	3	4	2	7	5	0	4	4
REW	0	4	3	2	3	3	6	0
EOP	3	2	5	0	6	0	8	0
FairSMOTE	4	5	3	2	3	2	10	0
MAAT	4	4	3	5	5	4	9	1
FairMask	0	7	1	2	3	1	6	0
MirrorFair	2	8	4	4	6	1	12	0
Overall	16	34	21	22	31	11	55	5

Second, these bias mitigation methods generally increase the SR, TPR, and FPR for the two least privileged groups ($Group_4$ and $Group_3$). Particularly, SR, TPR, and FPR for $Group_4$ are significantly increased in 67.9% (57/84), 51.2% (43/84), and 65.5% (55/84) of tasks, respectively. Across all eight methods studied, SR, TPR, and FPR for $Group_4$ are significantly increased in more scenarios than they are decreased.

This finding contrasts with previous research [25], which finds that bias mitigation methods in CV and NLP typically improve fairness by disadvantaging the most privileged group without benefiting the least privileged group when multiple sensitive attributes are considered.

Third, for the second privileged group ($Group_2$), the effects of bias mitigation methods are more balanced, with comparable increases and decreases in performance. For instance, SR is significantly increased in 23 tasks and decreased in 25 tasks. However, individual methods show varying trends; for example, FairMOTE is more likely to increase SR for $Group_2$, while MAAT tends to decrease it.

Finding 4: The zero-sum pattern of bias mitigation methods for tabular data also applies to multiple-attribute tasks. Unlike previous findings in CV and NLP, where methods often improve fairness by disadvantaging the most privileged group without benefiting the least privileged group in multiple-attributes tasks, our analysis reveals that methods for tabular data tend to decrease the selection rate, true positive rate, and false positive rate for the most privileged group, while increasing these metrics for the least privileged group.

7 RQ4: Bias Mitigation Without Harming Any Group

Our investigation of the previous RQs shows that existing bias mitigation methods for tabular data often benefit the unprivileged group but at a significant cost to the privileged group. In this RQ, we explore whether it is possible to enhance the benefits for the unprivileged group without negatively impacting the privileged group. A straightforward approach is to apply bias mitigation methods exclusively to the unprivileged group while retaining the original predictions for the privileged group, given that current methods primarily benefit unprivileged groups. We analyze the feasibility of this approach across both single-attribute tasks (**RQ4.1**) and multiple-attribute tasks (**RQ4.2**).

7.1 RQ4.1: Without Harming Any Group in Single-Attribute Tasks

Methodology. To implement this approach, we apply MirrorFair solely to the unprivileged group (denoted as *MirrorFairU*), as MirrorFair has been shown to outperform other methods and is considered state-of-the-art [35]. For comparison, we also implement a more aggressive approach aimed at equalizing the probability of receiving favorable outcomes (i.e., an equal selection rate) between the unprivileged and privileged groups. The implementation proceeds as follows: We randomly select 20% of the training data as a validation set and use the remaining 80% to train the model. The model is then used to predict labels for the privileged group members and the probability of receiving a favorable outcome for the unprivileged group members in the validation set. Based on these predictions, we calculate the selection rate for the privileged group, denoted as $x\%$. Next, we identify the top $x\%$ of predicted probabilities in the unprivileged group, with the lowest probability in this range set as the threshold for the unprivileged group. In the test data, if an unprivileged instance's predicted probability exceeds this threshold, it is assigned a favorable outcome; otherwise, it receives an unfavorable outcome. For privileged group instances, we retain the original model predictions. This method ensures that both groups achieve the same selection rate in the test data, and we refer to it as *Naivebase*.

Since MirrorFair is the state-of-the-art bias mitigation method for tabular data, we compare it against MirrorFairU and NaiveBase. In addition to evaluating fairness and group-specific performance, we also assess overall ML performance for a comprehensive evaluation. ML performance is a critical functional requirement of ML software. To measure it, we follow previous studies [12, 14, 35] and use a set of five common metrics: accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC). For precision, recall, and F1-score, we report macro-average values, as done in prior research [12, 14, 35], to provide an overall comparison across favorable and unfavorable classes by averaging the results for both classes. MCC is chosen for its ability to handle imbalanced class distributions, which are prevalent in fairness research benchmark datasets [12, 14]. This addresses concerns that accuracy, though most widely used in fairness studies [13], may not adequately reflect performance in the presence of imbalanced class distributions [12, 14].

To conduct this comparison, we use a win-tie-loss analysis and apply the Mann-Whitney U-test described in Section 4 to ensure the statistical significance of the results. For accuracy, precision, recall, F1-score, MCC, SR, and TPR, higher values indicate better performance; for FPR, SPD, EOD, and AOD, lower values reflect better performance and fairness. We compare the three methods across 32 single-attribute tasks. In each task, if MirrorFairU or NaiveBase produces significantly better results than MirrorFair, it is labeled as a "Win." If the results are significantly worse, it is marked as a "Loss." If there is no statistically significant difference, the outcome is classified as a "Tie." We then count the number of tasks where MirrorFairU or NaiveBase achieves a Win, Tie, or Loss.

Additionally, in certain tasks, such as recidivism prediction, overall selection rates may not be limited. However, in resource-constrained applications such as loan approvals, selection rates might

Table 5. (RQ4.1) Comparative analysis of MirrorFairU vs. MirrorFair and NaiveBase vs. MirrorFair across 32 single-attribute tasks. The win-tie-loss analysis shows that MirrorFairU improves SR_P, TPR_P, and FPR_P compared to MirrorFair, while preserving SR_U, TPR_U, FPR_U and achieving similar or better overall ML performance, though with reduced fairness. In contrast, NaiveBase offers greater benefits to the unprivileged group, but at the cost of significantly lower overall ML performance.

Metric	MirrorFairU			NaiveBase		
	Win	Tie	Loss	Win	Tie	Loss
SR_P	19	12	1	19	12	1
TPR_P	16	15	1	17	14	1
FPR_P	0	14	18	0	13	19
SR_U	1	31	0	24	6	2
TPR_U	1	31	0	22	9	1
FPR_U	0	31	1	1	7	24
Accuracy	2	28	2	0	8	24
Recall	7	22	3	12	8	12
Precision	0	26	6	1	13	18
F1-score	7	22	3	7	9	16
MCC	7	22	3	4	14	14
SPD	1	15	16	19	9	4
EOD	2	23	7	1	10	21
AOD	1	19	12	2	12	18

be capped. Thus, we also compare the overall selection rates of these approaches to evaluate if they significantly increase resource demands.

Results. Table 5 presents the results of the win-tie-loss analysis. We then compare MirrorFair with MirrorFairU and NaiveBase, respectively.

MirrorFairU vs. MirrorFair: MirrorFairU breaks the zero-sum pattern by enhancing benefits for the unprivileged group without reducing those for the privileged group. Specifically, MirrorFairU achieves higher SP_P and TPR_P compared to MirrorFair, as it applies bias mitigation only to the unprivileged group. Both methods achieve similar performance for the unprivileged group. Despite both methods treating the unprivileged group the same way, they do not yield identical results across all 32 tasks. This is due to the inherent nondeterminism in ML, where repeated training runs can produce different outcomes [32]. However, this nondeterminism has an impact on only one task, and does not affect our main findings.

In terms of ML performance, the win-tie-loss analysis reveals that MirrorFairU and MirrorFair achieve similar accuracy levels, and MirrorFairU outperforms MirrorFair in recall, F1-score, and MCC. However, MirrorFairU has lower precision, which is expected since recall and precision are often conflicting objectives [8]. This is why researchers often rely on the F1-score, which balances these two metrics. Overall, MirrorFairU surpasses MirrorFair in F1-score, with significantly higher results in 7 tasks and lower results in 3 tasks.

Regarding fairness metrics, MirrorFairU underperforms compared to MirrorFair on SPD, EOD, and AOD. This is because MirrorFairU increases benefits for the unprivileged group to match those of MirrorFair without reducing the benefits for the privileged group. However, from a welfare economics perspective, MirrorFairU achieves a Pareto improvement, meaning it makes at least one group better off without making others worse off [17]. This result exposes a limitation in existing fairness metrics: they primarily focus on the relative performance gap between groups and fail to account for cases where one group's benefits improve without negatively affecting the other. As a result, more sophisticated evaluation methods are needed to better balance fairness and group performance in these complex scenarios.

In addition, the notable number of ties in Table 5 is expected, as our goal in RQ4 is to show that applying MirrorFair exclusively to unprivileged groups (i.e., MirrorFairU) increases benefits for privileged groups, while preserving those for unprivileged groups and maintaining comparable

overall ML performance, relative to applying it to the entire population. As a result, ties are prevalent in metrics such as SR_U, TPR_U, and FPR_U, and overall ML performance metrics.
NaiveBase vs. MirrorFair: As shown in Table 5, NaiveBase provides greater benefits for both privileged and unprivileged groups, with higher SP_P in 19 tasks and higher SP_U in 24 tasks. However, this advantage comes with a significant cost: NaiveBase exhibits considerably lower accuracy than MirrorFair in 24 out of 32 tasks. Given the critical importance of ML performance for practical applications, NaiveBase's reduced accuracy makes it less feasible for real-world use. Therefore, we do not consider it further in our analysis.

Based on our analysis, MirrorFairU shows promise as a bias mitigation method. To further assess its practicality, we calculate its overall selection rate to ensure it does not impose excessive resource demands. We compare MirrorFairU to MirrorFair, the current state-of-the-art method for bias mitigation in tabular data. Our findings indicate that MirrorFairU leads to an average increase of 0.033 in the overall selection rate across all 32 tasks, compared to 0.023 for MirrorFair. This translates to a marginal increase of about 1% in resource requirements when using MirrorFairU instead of MirrorFair. Moreover, in tasks such as recidivism prediction, this increase in the selection rate does not demand additional social resources.

> **Finding 5:** Applying the state-of-the-art bias mitigation method, MirrorFair, exclusively to the unprivileged group breaks the zero-sum pattern by increasing benefits for the unprivileged group without diminishing those for the privileged group. This approach not only enhances fairness and improves outcomes for the unprivileged group, but also preserves the benefits for the privileged group while maintaining similar or even better overall ML performance compared to applying MirrorFair to the entire population. Additionally, this approach leads to an average increase of only 0.01 in the overall selection rate.

7.2 RQ4.2: Without Harming Any Group in Multiple-Attribute Tasks

Methodology. To further assess the applicability of MirrorFairU, we apply it to 12 multiple-attribute tasks. As shown in Table 4, MirrorFair primarily negatively impacts $Group_1$ and $Group_2$ while benefiting $Group_3$ and $Group_4$. Therefore, for the multiple-attribute tasks, we implement MirrorFairU by applying MirrorFair exclusively to $Group_3$ and $Group_4$. We then evaluate MirrorFairU's fairness, performance, and overall selection rates using the same methodology as in RQ4.1.
Results. Table 6 presents the results, which generally align with our findings in RQ4.1. MirrorFairU achieves comparable performance for $Group_3$ and $Group_4$ as MirrorFair, while improving the SR, TPR, and FPR for $Group_1$ and $Group_2$, as expected, since MirrorFair is applied only to $Group_1$ and $Group_2$. In terms of overall performance, MirrorFairU exhibits similar accuracy, F1-score, and MCC compared to MirrorFair. Moreover, as noted in RQ4.1, MirrorFairU shows better recall but worse precision than MirrorFair. Additionally, consistent with our findings in RQ4.1, MirrorFairU demonstrates poorer fairness than MirrorFair regarding SPD, EOD, and AOD.

We also calculate the overall selection rate for both MirrorFairU and MirrorFair. MirrorFairU shows an average increase of 0.081 across all 12 multiple-attribute tasks, compared to 0.064 for MirrorFair, resulting in a difference of just 0.017.

> **Finding 6:** Applying the state-of-the-art bias mitigation method, MirrorFair, exclusively to unprivileged groups in multi-attribute tasks increases benefits for privileged groups compared to the standard MirrorFair, while preserving benefits for unprivileged groups and maintaining

Table 6. (RQ4.2) Comparative analysis of MirrorFairU vs. MirrorFair across 12 multiple-attribute tasks. The win-tie-loss results indicate that MirrorFairU enhances benefits for $Group_1$ and $Group_2$ compared to MirrorFair, while maintaining the benefits for $Group_3$ and $Group_4$ and achieving similar overall ML performance.

Metric	MirrorFairU			Metric	MirrorFairU		
	Win	Tie	Loss		Win	Tie	Loss
SR_1	8	4	0	Accuracy	2	9	1
TPR_1	8	4	0	Recall	3	8	1
FPR_1	0	3	9	Precision	0	8	4
SR_2	6	5	1	F1-score	2	8	2
TPR_2	4	7	1	MCC	2	9	1
FPR_2	1	6	5	SPD	1	5	6
SR_3	0	12	0	EOD	0	10	2
TPR_3	0	12	0	AOD	1	6	5
FPR_3	0	12	0				
SR_4	0	12	0				
TPR_4	0	12	0				
FPR_4	0	12	0				

similar ML performance. Additionally, the average difference in the overall selection rate between the two approaches is only 0.017.

8 Discussion

This section explores the potential reason for the difference between our findings and prior research. It also discusses the qualitative insights into the observed zero-sum trade-offs, the implications for various stakeholders, and potential threats to validity.

8.1 Comparison With Previous Work

Previous studies in CV and NLP have found that bias mitigation methods in these fields often lead to a leveling-down effect, where the performance of all groups is degraded [25, 43]. In contrast, our findings reveal that bias mitigation methods applied to tabular data exhibit a zero-sum pattern. In this case, the methods improve outcomes for the unprivileged group while reducing them for the privileged group. This pattern holds consistently across all eight bias mitigation methods we examine. In this section, we explore the possible reason for this difference.

Zietlow et al. [43] suggest that the leveling-down phenomenon in CV occurs due to the high-dimensional data and high-capacity models often used. These models typically achieve near-zero training error when fairness constraints are applied, but struggle to generalize to new test data, leading to degraded performance for all groups.

In our experiments, however, we observe an average training error of 9% when applying the eight bias mitigation methods across 32 single-attribute tasks. Additionally, the performance impact for each group on the training data is consistent with the test data. Specifically, these methods increase the SR, TPR, and FPR for the unprivileged group while decreasing them for the privileged group. Due to space constraints, Table 7 presents only the SR metric, but similar patterns hold across other metrics. We find that all eight methods tend to decrease SR_P and increase SR_U on the training data. This suggests that bias mitigation methods for tabular data may generalize better from training to test data compared to those used in CV and NLP, leading to the observed differences.

8.2 Qualitative Insights

We provide qualitative insights into the mechanisms of each bias mitigation method to explain why they lead to observed trade-offs between privileged and unprivileged groups. **(1) ADV** reduces

Table 7. Number of tasks where each bias mitigation method significantly increases (↑), decreases (↓), or does not significantly impact (−) the SR for each group in the training data. Overall, when applying bias mitigation methods for tabular data to the training data, these methods tend to decrease SR for the privileged group while increasing it for the unprivileged group.

Method	SR_P (↑)	SR_P (−)	SR_P (↓)	SR_U (↑)	SR_U (−)	SR_U (↓)
ADV	6	9	17	13	13	6
REW	0	7	25	24	8	0
EOP	1	12	19	22	10	0
FairSMOTE	8	9	15	24	4	4
LTDD	1	15	16	18	14	0
MAAT	2	9	21	27	5	0
FairMask	0	18	14	12	20	0
MirrorFair	4	11	17	29	1	2
Overall	22	90	144	169	75	12

the model's reliance on sensitive attributes and their proxies, suppressing correlations that disproportionately benefit privileged groups. This limits the predictive advantage of features tied to privilege, decreasing their benefits. Unprivileged groups gain as the model learns to rely on features more equitably distributed across demographics, improving their outcomes. **(2) REW** adjusts instance weights to prioritize unprivileged groups, focusing the model's learning on improving their outcomes. This rebalancing reduces the influence of privileged group instances, decreasing their benefits while enhancing those of unprivileged groups. **(3) EOP** adjusts predictions to equalize false positive and false negative rates across groups. These adjustments often reduce accuracy for privileged groups by correcting biases that previously favor them. Unprivileged groups benefit as their predictive outcomes become more equitable. **(4) FairSMOTE** generates synthetic data for unprivileged groups, increasing their representation in training. This improves the model's ability to generalize for unprivileged groups but reduces the relative influence of privileged group data, decreasing their benefits. **(5) LTDD** removes biased components of training data that favor privileged groups, reducing the model's reliance on features tied to privilege. This levels the playing field, improving outcomes for unprivileged groups but decreasing benefits for privileged ones. **(6) MAAT** combines predictions from models optimized separately for fairness and performance. The fairness-optimized model redistributes benefits to reduce disparities, often at the cost of benefits for privileged groups. Unprivileged groups gain as their outcomes are specifically targeted for improvement. **(7) FairMask** predicts sensitive attributes from other features and relabels them to reduce bias. By masking biased correlations that favor privileged groups, the model redistributes benefits, improving outcomes for unprivileged groups while reducing those of privileged groups. **(8) MirrorFair** uses counterfactual datasets to balance predictions, reducing the model's reliance on patterns that favor privileged groups. This adjustment corrects disadvantages for unprivileged groups but decreases the benefits for privileged ones as outcomes are redistributed.

8.3 Implications

For SE researchers: **(1) Research reproducibility.** Fairness research findings in CV and NLP may not generalize to tabular data, as shown in our analysis (RQ1 and RQ3). This underscores the need for domain-specific evaluations of bias mitigation methods. SE researchers should replicate and validate software fairness studies across data types and task domains to identify limitations and ensure robust outcomes in diverse ML applications. **(2) Comprehensive evaluation practices.** Current software fairness research primarily focuses on overall performance and fairness, often overlooking the performance of individual demographic groups. Our findings (RQs1–3) show significant trade-offs in group benefits, which could hinder the practical adoption of bias mitigation methods. SE researchers should adopt granular evaluation practices that assess impacts on each

demographic group, highlighting trade-offs and enabling more effective interventions that balance fairness and group-specific outcomes. **(3) Rethinking fairness metrics.** Our findings (RQ4) show that while MirrorFairU provides comparable benefits for unprivileged groups and better outcomes for privileged groups, existing fairness metrics still favor MirrorFair by focusing on relative disparities and often overlooking the absolute performance of each group. This limitation can obscure opportunities to improve outcomes for all groups. Fairness metrics should evolve to balance equity across groups with absolute benefits for each, considering both relative disparities and individual group outcomes to promote fairness and well-being for all demographics. **(4) Innovative bias mitigation methods.** The trade-offs between groups in existing methods present challenges in high-stakes applications like criminal justice, healthcare, and finance, where the performance of all groups must remain high. Our results (RQ4) demonstrate the feasibility of achieving fairness without such trade-offs, providing a viable path forward. SE researchers should develop bias mitigation methods that optimize both relative fairness and absolute performance across groups. Prioritizing these solutions can lead to interventions that are ethical, effective, and more likely to be adopted in real-world systems.

For software engineers and practitioners: **(1) Mediating trade-offs and addressing conflicting requirements.** Anti-discrimination laws require software engineers and practitioners to ensure fairness in ML systems. However, our findings (RQs1–3) reveal that fairness interventions often involve trade-offs, where improvements for unprivileged groups significantly reduce outcomes for privileged groups, especially as fairness gains increase. These trade-offs can hinder the adoption of software systems. Engineers and practitioners should consider these conflicting requirements in the SE process, using negotiation, mediation, conflict resolution, and multi-objective optimization strategies to balance these requirements. **(2) Leveraging evidence-based practices.** Our evaluation of eight advanced bias mitigation methods (RQ1 and RQ3) highlights their impacts on each group's performance across single and multiple sensitive attribute scenarios, providing guidance for selecting methods aligned with specific group performance goals. Additionally, our results (RQ4) propose a promising approach: selectively applying bias mitigation to enhance fairness and unprivileged group outcomes while preserving privileged group benefits. This strategy offers a practical path for achieving balanced outcomes, enabling engineers and practitioners to develop fair and effective ML systems.

For policymakers: **(1) Establishing fairness standards linked to group-specific performance.** Current regulations aim to promote fairness in software systems, yet our findings (RQ1 and RQ3) show that bias mitigation methods often improve fairness at the expense of privileged groups, with greater fairness gains correlating with larger reductions in their outcomes (RQ2). These trade-offs could hinder the adoption of fairness policies. Policymakers should establish standards requiring detailed reporting of fairness metrics along with their impact on all demographic groups, encouraging balanced approaches that consider the broader implications of fairness interventions. **(2) Challenging the zero-sum perception of fairness.** Our findings (RQ4) demonstrate that fairness improvements for underprivileged groups do not always demand significant sacrifices from privileged groups. Policymakers should counter the misconception of fairness as a zero-sum game through awareness campaigns and stakeholder dialogues. Presenting evidence that fairness can be achieved without severe trade-offs will build trust and support for fairness-focused regulations and fair software systems.

8.4 Threats to Validity

Selection of bias mitigation tasks. The choice of tasks may pose a threat to validity. To address this, we employ the same set of 44 bias mitigation tasks from a recent study [35], covering five widely used datasets and four commonly applied ML models. These datasets reflect real-world applications

across various domains and include sensitive attributes frequently considered in practice. Analyzing SE-specific datasets could further strengthen the relevance of this work. However, we are unaware of any publicly available SE tabular datasets suitable for such fairness studies. This may explain why recent related papers [4, 9, 12–14, 21, 24, 31, 35, 39] also rely on datasets from other domains.
Selection of bias mitigation methods. Given the extensive volume of research on bias mitigation [20], it is challenging to include all existing methods in this study. To mitigate this threat, we select a set of eight representative bias mitigation methods, encompassing both the most widely used methods and recently proposed state-of-the-art techniques.
Selection of statistical analysis methods. We employ the Mann-Whitney U-test, Cliff's δ, and Spearman's correlation coefficient for our statistical analysis. These methods are commonly used in the SE literature [13, 14]. They do not rely on the assumption of data normality, making them appropriate for our study, which involves various data that may not conform to a normal distribution.
Selection of metrics. To mitigate the potential threat related to metric selection, we adopt the same set of metrics used in recent software fairness studies [12, 14, 35]. This includes three widely recognized fairness metrics and five commonly used ML performance metrics.
Implementation of bias mitigation methods. To address this threat, we directly use the code provided by the original authors of the bias mitigation methods, ensuring implementation reliability. Additionally, in alignment with prior work [14], each method is applied to each task 20 times to mitigate the influence of randomness. All code and datasets used in this study have been made available in an open repository [1] to facilitate replication and validation.

9 Conclusion

This paper presents a comprehensive study on the impact of existing bias mitigation methods for tabular data, analyzing their effects on the model's performance for different demographic groups. We observe that all methods lead to a zero-sum trade-off, where improvements for unprivileged groups are associated with reduced outcomes for privileged groups. Given that the perception of such trade-offs might impede the broader adoption of fairness-focused policies, we explore the application of the state-of-the-art bias mitigation method exclusively to unprivileged groups. Our preliminary findings suggest that this approach can enhance benefits for unprivileged groups without compromising outcomes for privileged groups, and it maintains overall ML performance (e.g., accuracy and F1-score) comparable to existing methods. Building on these insights, we plan to develop more effective bias mitigation strategies that avoid the zero-sum trade-off, thereby facilitating the broader adoption of bias mitigation methods in ML software.

10 Data Availability

We have made the replication package publicly available in a repository [1], which includes all datasets, code, and intermediate results from our study.

Acknowledgments

This research is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No. AISG2-RP-2020-019); by the National Research Foundation Singapore and the Cyber Security Agency of Singapore under the National Cybersecurity R&D Programme (NCRP25-P04-TAICeN); and by the National Research Foundation, Prime Minister's Office, Singapore under the Campus for Research Excellence and Technological Enterprise (CREATE) programme. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not reflect the views of the National Research Foundation Singapore or the Cyber Security Agency of Singapore.

References

[1] 2025. Replication package. https://github.com/chenzhenpeng18/FSE25-ZeroSum.

[2] Lauren Alvarez and Tim Menzies. 2023. Don't lie to me: Avoiding malicious explanations with STEALTH. *IEEE Softwaware* 40, 3 (2023), 43–53.

[3] Kwabena Ebo Bennin, Jacky Keung, Akito Monden, Passakorn Phannachitta, and Solomon Mensah. 2017. The significant effects of data sampling approaches on software defect prioritization and classification. In *Proceedings of the 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2017*. 364–373.

[4] Sumon Biswas and Hridesh Rajan. 2020. Do the machine learning models on a crowd sourced platform exhibit bias? An empirical study on model fairness. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2020*. 642–653.

[5] Sumon Biswas and Hridesh Rajan. 2021. Fair preprocessing: Towards understanding compositional fairness of data transformers in machine learning pipeline. In *Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2021*. 981–993.

[6] N Derek Brown, Drew S Jacoby-Senghor, and Isaac Raymundo. 2022. If you rise, I fall: Equality is prevented by the misperception that it harms advantaged groups. *Science Advances* 8, 18 (2022), eabm2385.

[7] Yuriy Brun and Alexandra Meliou. 2018. Software fairness. In *Proceedings of the 2018 ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018*. 754–759.

[8] Michael Buckland and Fredric Gey. 1994. The relationship between recall and precision. *Journal of the American Society for Information Science* 45, 1 (1994), 12–19.

[9] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in machine learning software: Why? How? What to do?. In *Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2021*. 429–440.

[10] Joymallya Chakraborty, Suvodeep Majumder, Zhe Yu, and Tim Menzies. 2020. Fairway: A way to build fair ML software. In *Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2020*. 654–665.

[11] Zhenpeng Chen, Jie M. Zhang, Max Hort, Mark Harman, and Federica Sarro. 2024. Fairness testing: A comprehensive survey and analysis of trends. *ACM Transactions on Software Engineering and Methodology* 33, 5 (2024), 137:1–137:59.

[12] Zhenpeng Chen, Jie M. Zhang, Federica Sarro, and Mark Harman. 2022. MAAT: A novel ensemble approach to addressing fairness and performance bugs for machine learning software. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022*. 1122–1134.

[13] Zhenpeng Chen, Jie M. Zhang, Federica Sarro, and Mark Harman. 2023. A comprehensive empirical study of bias mitigation methods for machine learning classifiers. *ACM Transactions on Software Engineering and Methodology* 32, 4 (2023), 106:1–106:30.

[14] Zhenpeng Chen, Jie M. Zhang, Federica Sarro, and Mark Harman. 2024. Fairness improvement with multiple protected attributes: How far are we?. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, ICSE 2024*. 160:1–160:13.

[15] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT 2019*. 329–338.

[16] Xuanqi Gao, Juan Zhai, Shiqing Ma, Chao Shen, Yufei Chen, and Qian Wang. 2022. Fairneuron: Improving deep neural network fairness with adversary games on selective neurons. In *Proceedings of the 44th IEEE/ACM International Conference on Software Engineering, ICSE 2022*. 921–933.

[17] Sven Ove Hansson. 2004. Welfare, justice, and Pareto efficiency. *Ethical Theory and Moral Practice* 7 (2004), 361–380.

[18] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems 2016, NIPS 2016*. 3315–3323.

[19] Jennifer Horkoff. 2019. Non-Functional requirements for machine learning: Challenges and new directions. In *Proceedings of the 27th IEEE International Requirements Engineering Conference, RE 2019*. 386–391.

[20] Max Hort, Zhenpeng Chen, Jie M. Zhang, Mark Harman, and Federica Sarro. 2024. Bias mitigation for machine learning classifiers: A comprehensive survey. *ACM Journal on Responsible Computing* 1, 2 (2024), 11:1–11:52.

[21] Max Hort, Jie M. Zhang, Federica Sarro, and Mark Harman. 2021. Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods. In *Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2021*. 994–1006.

[22] Faisal Kamiran and Toon Calders. 2011. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2011), 1–33.

[23] Barbara A. Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart M. Charters, Shirley Gibbs, and Amnart Pohthong. 2017. Robust statistical methods for empirical software engineering. *Empirical Software Engineering* 22, 2 (2017), 579–630.

[24] Yanhui Li, Linghan Meng, Lin Chen, Li Yu, Di Wu, Yuming Zhou, and Baowen Xu. 2022. Training data debugging for the fairness of machine learning software. In *Proceedings of the 44th IEEE/ACM International Conference on Software Engineering, ICSE 2022*. 2215–2227.
[25] Gaurav Maheshwari, Aurélien Bellet, Pascal Denis, and Mikaela Keller. 2023. Fair without leveling down: A new intersectional fairness definition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*. 9018–9032.
[26] Patrick E McKnight and Julius Najab. 2010. Mann-Whitney U test. *The Corsini Encyclopedia of Psychology* (2010), 1–1.
[27] Aditya Krishna Menon and Robert C. Williamson. 2018. The cost of fairness in binary classification. In *Proceedings of the Conference on Fairness, Accountability and Transparency, FAT 2018*. 107–118.
[28] Brent Mittelstadt, Sandra Wachter, and Chris Russell. 2023. The unfairness of fair machine learning: Levelling down and strict egalitarianism by default. *arXiv preprint arXiv:2302.02404* (2023).
[29] Jerome L Myers, Arnold D Well, and Robert F Lorch Jr. 2013. *Research design and statistical analysis*. Routledge.
[30] Michael I Norton and Samuel R Sommers. 2011. Whites see racism as a zero-sum game that they are now losing. *Perspectives on Psychological science* 6, 3 (2011), 215–218.
[31] Kewen Peng, Joymallya Chakraborty, and Tim Menzies. 2023. FairMask: Better fairness via model-based rebalancing of protected attributes. *IEEE Transactions on Software Engineering* 49, 4 (2023), 2426–2439.
[32] Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. 2020. Problems and opportunities in training deep learning software systems: an analysis of variance. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020*. 771–783.
[33] Saeid Tizpaz-Niari, Ashish Kumar, Gang Tan, and Ashutosh Trivedi. 2022. Fairness-aware configuration of machine learning libraries. In *Proceedings of the 44th IEEE/ACM International Conference on Software Engineering, ICSE 2022*. 909–920.
[34] Jonathan Wolff. 2001. Levelling down. In *Challenges to Democracy: Ideas, Involvement and Institutions*. Springer, 18–32.
[35] Ying Xiao, Jie M Zhang, Yepang Liu, Mohammad Reza Mousavi, Sicen Liu, and Dingyuan Xue. 2024. MirrorFair: Fixing fairness bugs in machine learning software via counterfactual predictions. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 2121–2143.
[36] Junjie Yang, Jiajun Jiang, Zeyu Sun, and Junjie Chen. 2024. A large-scale empirical study on improving the fairness of image classification models. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2024*. 210–222.
[37] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*. 1171–1180.
[38] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018*. 335–340.
[39] Jie M. Zhang and Mark Harman. 2021. Ignorance and prejudice in software fairness. In *Proceedings of the 43rd IEEE/ACM International Conference on Software Engineering, ICSE 2021*. 1436–1447.
[40] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. 2022. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering* 48, 2 (2022), 1–36.
[41] Mengdi Zhang and Jun Sun. 2022. Adaptive fairness improvement based on causality analysis. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022*. 6–17.
[42] Haibin Zheng, Zhiqing Chen, Tianyu Du, Xuhong Zhang, Yao Cheng, Shouling Ji, Jingyi Wang, Yue Yu, and Jinyin Chen. 2022. NeuronFair: Interpretable white-box fairness testing through biased neuron identification. In *Proceedings of the 44th IEEE/ACM International Conference on Software Engineering, ICSE 2022*. 1519–1531.
[43] Dominik Zietlow, Michael Lohaus, Guha Balakrishnan, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Chris Russell. 2022. Leveling down in computer vision: Pareto inefficiencies in fair deep classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*. 10400–10411.

Received 2024-09-12; accepted 2025-04-01